第二册｜《自我逻辑演化》
Volume II | Self-Logic Evolution

文章本天成，妙手偶得之。
The work is born of nature; the deft hand merely happens upon it.

本文为结构性模型呈现，不构成任何形式的指导或身份主张。 This text presents structural models and makes no claims of guidance or identity. 请基于内容本身判断其有效性。 Please evaluate it solely on the basis of its content.

------谢凯凡 kaifanxie

17｜自我不是实体
17 | The Self Is Not an Entity

自我不是一个固定对象。
The self is not a fixed object.
自我是系统对自身的模型。
The self is a system's model of itself.
模型会变化。
Models change.
因此，自我会变化。
Therefore, the self changes.

17.1 自我模型
17.1 Self-Model

自我模型用于预测自身行为。
The self-model is used to predict one's own behaviour.
它不是事实描述。
It is not a factual description.
它是控制工具。
It is a control tool.

17.2 自指
17.2 Self-Reference

系统可以将自身作为输入。
A system can take itself as input.
这称为自指。
This is called self-reference.
自指提高控制精度。
Self-reference increases control precision.
也增加不稳定性。
It also increases instability.

17.3 递归
17.3 Recursion

自我模型可以被再次建模。
A self-model can be modelled again.
递归增加表达能力。
Recursion increases expressive power.
递归也放大误差。
Recursion also amplifies error.

17.4 工程结论
17.4 Engineering Conclusion

自我是功能结构。

The self is a functional structure.
不是本体承诺。
It is not an ontological commitment.

18｜身份、角色与稳定叙事
18 | Identity, Roles, and Stable Narratives

身份不是你是谁。
Identity is not who you are.
身份是外部接口。
Identity is an external interface.

18.1 角色
18.1 Roles

角色是环境对系统的期望。
Roles are environmental expectations placed on a system.
角色简化交互。
Roles simplify interaction.
也限制行为空间。
They also constrain behavioural space.

18.2 身份锁定
18.2 Identity Lock-in

重复使用同一身份会导致锁定。
Repeated use of the same identity leads to lock-in.
锁定提高可预测性。
Lock-in increases predictability.
也降低可塑性。
It also reduces plasticity.

18.3 叙事稳定
18.3 Narrative Stability

系统通过叙事维持连续性。
Systems maintain continuity through narratives.
叙事压缩历史。
Narratives compress history.
压缩隐藏断裂。
Compression hides rupture.

18.4 工程结论
18.4 Engineering Conclusion

身份是工具。
Identity is a tool.
不是本质。
It is not essence.

19｜价值的演化
19 | Evolution of Values

价值不是先验的。
Values are not a priori.
价值从反馈中生成。
Values emerge from feedback.

19.1 价值更新
19.1 Value Update

价值随成功路径强化。
Values strengthen along successful paths.

失败路径削弱价值权重。
Failed paths weaken value weights.

19.2 价值冲突
19.2 Value Conflict
系统可同时持有冲突价值。
A system can hold conflicting values simultaneously.
冲突反映多目标控制。
Conflict reflects multi-objective control.

19.3 价值坍缩
19.3 Value Collapse
环境剧变会引发价值坍缩。
Environmental shocks can trigger value collapse.
旧排序失效。
Old rankings fail.
系统进入无序态。
The system enters a disordered state.

19.4 工程结论
19.4 Engineering Conclusion
价值是动态排序函数。
Values are dynamic ranking functions.
不是道德命题。
They are not moral propositions.
20｜自我控制与内在冲突
20 | Self-Control and Internal Conflict
系统内部并非单一目标。
A system is not internally single-goal.
内部冲突是常态。
Internal conflict is the norm.

20.1 多子系统
20.1 Multiple Subsystems
系统由多个子系统构成。
A system is composed of multiple subsystems.
子系统具有不同目标。
Subsystems have different goals.

20.2 冲突调解
20.2 Conflict Mediation
自我控制是冲突调解机制。
Self-control is a conflict mediation mechanism.
它不是压制。
It is not suppression.
它是调度。
It is scheduling.

20.3 失败模式
20.3 Failure Modes
调解失败导致行为失序。
Mediation failure leads to behavioural disorder.
失序不是邪恶。
Disorder is not evil.

失序是控制失效。
Disorder is control failure.

20.4 工程结论
20.4 Engineering Conclusion

自我控制是工程问题。
Self-control is an engineering problem.
不是意志品质。
It is not a moral quality.

21｜意识作为监控层
21 | Consciousness as a Monitoring Layer

意识不是决策中心。
Consciousness is not the decision centre.
意识是系统的监控层。
Consciousness is the monitoring layer of the system.

21.1 监控而非控制
21.1 Monitoring, Not Control

意识并不直接产生行为。
Consciousness does not directly generate behaviour.
行为在意识之前已被触发。
Behaviour is triggered before consciousness.
意识记录、解释并评估。
Consciousness records, interprets, and evaluates.

21.2 可报告性
21.2 Reportability

能被意识捕获的内容才能被报告。
Only contents captured by consciousness can be reported.
不可报告不等于不存在。
Unreportable does not mean non-existent.
多数运算发生在监控层之外。
Most computation occurs outside the monitoring layer.

21.3 意识延迟
21.3 Conscious Delay

意识存在系统性延迟。
Consciousness has systematic delay.
延迟导致事后合理化。
Delay leads to post-hoc rationalisation.
解释并非因果源头。
Explanations are not causal origins.

21.4 工程结论
21.4 Engineering Conclusion

意识提高可调试性。
Consciousness increases debuggability.
但不提高控制精度。
But it does not increase control precision.

22｜自由意志作为接口幻觉
22 | Free Will as an Interface Illusion

自由意志不是物理变量。
Free will is not a physical variable.
自由意志是主观接口体验。

Free will is a subjective interface experience.

22.1 选择感

22.1 Sense of Choice

选择感源于多路径并存。

The sense of choice arises from the coexistence of multiple paths.

当路径被裁剪，选择感消失。

When paths are pruned, the sense of choice disappears.

22.2 可控感

22.2 Sense of Control

可控感不等于可控性。

The sense of control is not equivalent to controllability.

可控感是预测成功的副产品。

The sense of control is a by-product of successful prediction.

22.3 工程结论

22.3 Engineering Conclusion

自由意志不是必要假设。

Free will is not a necessary assumption.

系统可在无自由意志下分析。

Systems can be analysed without free will.

23｜责任、归因与叙事修复

23 | Responsibility, Attribution, and Narrative Repair

系统必须解释自身行为。

Systems must explain their own behaviour.

解释用于维持结构稳定。

Explanations are used to maintain structural stability.

23.1 归因机制

23.1 Attribution Mechanisms

归因是因果压缩。

Attribution is causal compression.

归因减少不确定性。

Attribution reduces uncertainty.

也可能引入偏差。

It can also introduce bias.

23.2 责任叙事

23.2 Responsibility Narratives

责任叙事用于维持社会接口。

Responsibility narratives maintain social interfaces.

叙事不等于真实因果。

Narratives are not equal to true causality.

23.3 叙事修复

23.3 Narrative Repair

当叙事破裂，系统感到不稳定。

When narratives break, systems feel unstable.

修复叙事可恢复连续性。

Repairing narratives restores continuity.

但可能掩盖结构问题。

But it may conceal structural problems.

23.4 工程结论

23.4 Engineering Conclusion

归因是必要机制。

Attribution is a necessary mechanism.

但不能替代结构分析。

But it cannot replace structural analysis.

24｜自我一致性与内在神话

24 | Self-Consistency and Internal Myth

系统倾向维持自我一致。

Systems tend to maintain self-consistency.

一致性降低计算成本。

Consistency reduces computational cost.

24.1 内在神话

24.1 Internal Myth

内在神话是高度压缩的自我解释。

Internal myths are highly compressed self-explanations.

神话增强稳定性。

Myths enhance stability.

也限制更新。

They also limit updating.

24.2 防御性一致

24.2 Defensive Consistency

系统会抵抗破坏一致性的证据。

Systems resist evidence that disrupts consistency.

这是保护机制。

This is a protective mechanism.

24.3 神话破裂

24.3 Myth Rupture

当神话破裂，系统进入高不确定性。

When myths rupture, systems enter high uncertainty.

这是重组前兆。

This is a precursor to reorganisation.

24.4 工程结论

24.4 Engineering Conclusion

一致性不是美德。

Consistency is not a virtue.

它是成本优化策略。

It is a cost-optimisation strategy.

25｜极限状态与系统保护

25 | Extreme States and System Protection

系统在极端条件下会改变规则。

Systems change rules under extreme conditions.

这是生存机制。

This is a survival mechanism.

25.1 应急模式

25.1 Emergency Modes

应急模式牺牲精度换取速度。

Emergency modes sacrifice precision for speed.

长期运行会损伤系统。

Long-term operation damages the system.

25.2 保护性失真
25.2 Protective Distortion

系统可能主动失真感知。
Systems may actively distort perception.
这是保护而非错误。
This is protection, not error.

25.3 恢复
25.3 Recovery

极限状态后必须恢复正常规则。
After extreme states, normal rules must be restored.
否则系统会固化。
Otherwise, the system hardens.

25.4 工程结论
25.4 Engineering Conclusion

极限状态不是常态。
Extreme states are not normal.
保护机制需要退出条件。
Protective mechanisms require exit conditions.

26｜自我演化的边界条件
26 | Boundary Conditions of Self-Evolution

演化不是无限的。
Evolution is not unlimited.
演化受边界条件约束。
Evolution is constrained by boundary conditions.

26.1 结构不可逾越性
26.1 Structural Non-Transgressibility

并非所有结构都可被演化跨越。
Not all structures can be crossed by evolution.
某些限制是硬约束。
Some limits are hard constraints.
硬约束无法通过学习消除。
Hard constraints cannot be removed by learning.

26.2 能量与时间上限
26.2 Energy and Time Limits

演化消耗能量。
Evolution consumes energy.
演化消耗时间。
Evolution consumes time.
资源上限决定演化深度。
Resource limits determine evolutionary depth.

26.3 不可逆损伤
26.3 Irreversible Damage

某些错误造成不可逆损伤。
Some errors cause irreversible damage.
不可逆性定义风险边界。
Irreversibility defines risk boundaries.
系统必须提前规避。

Systems must avoid them in advance.

26.4 工程结论

26.4 Engineering Conclusion

自我演化必须尊重边界。

Self-evolution must respect boundaries.

突破边界不是成长。

Breaking boundaries is not growth.

27｜多重自我与上下文切换

27 | Multiple Selves and Context Switching

系统可表现为多个自我。

A system can manifest multiple selves.

多重自我是功能分化。

Multiple selves are functional differentiation.

27.1 上下文自我

27.1 Contextual Selves

不同环境激活不同子系统。

Different environments activate different subsystems.

每个子系统形成局部自我。

Each subsystem forms a local self.

27.2 切换成本

27.2 Switching Costs

上下文切换存在成本。

Context switching has costs.

频繁切换降低整体效率。

Frequent switching reduces overall efficiency.

27.3 冲突管理

27.3 Conflict Management

多重自我可能冲突。

Multiple selves may conflict.

冲突需要调度而非统一。

Conflicts require scheduling, not unification.

27.4 工程结论

27.4 Engineering Conclusion

多重自我是优势。

Multiple selves are an advantage.

但需控制切换成本。

But switching costs must be controlled.

28｜自我叙事的生成与崩解

28 | Generation and Breakdown of Self-Narratives

自我叙事不是事实记录。

Self-narratives are not factual records.

自我叙事是压缩历史的工具。

Self-narratives are tools for compressing history.

28.1 连续性的构造

28.1 Construction of Continuity

叙事创造连续感。

Narratives create a sense of continuity.

连续感降低不确定性。

Continuity reduces uncertainty.

28.2 断裂的掩盖

28.2 Concealment of Rupture

叙事会掩盖断裂。

Narratives conceal ruptures.

掩盖提高稳定性。

Concealment increases stability.

28.3 叙事崩解

28.3 Narrative Breakdown

当压缩失败，叙事崩解。

When compression fails, narratives break down.

系统暴露于真实复杂性。

The system is exposed to real complexity.

28.4 工程结论

28.4 Engineering Conclusion

叙事是必要工具。

Narratives are necessary tools.

但不能被误认为现实本身。

But they must not be mistaken for reality itself.

29｜自我优化的陷阱

29 | Traps of Self-Optimisation

优化并非总是正向。

Optimisation is not always positive.

局部优化可能破坏整体。

Local optimisation may damage the whole.

29.1 单指标陷阱

29.1 Single-Metric Trap

单一指标易于控制。

Single metrics are easy to control.

但易被操纵。

But they are easy to game.

29.2 内耗

29.2 Internal Friction

过度优化引发内耗。

Over-optimisation causes internal friction.

内耗消耗资源却无外部收益。

Friction consumes resources without external gain.

29.3 停止条件

29.3 Stopping Conditions

优化需要停止条件。

Optimisation requires stopping conditions.

没有停止条件的优化是病态的。

Optimisation without stopping conditions is pathological.

29.4 工程结论

29.4 Engineering Conclusion

自我优化必须受控。

Self-optimisation must be controlled.

否则系统将自我侵蚀。
Otherwise, the system will self-erode.

30｜完成态、冻结与版本切换

30 | Completion State, Freezing, and Version Switching

系统并非持续演化。
Systems do not evolve continuously.
某些阶段需要冻结。
Some phases require freezing.

30.1 完成态

30.1 Completion State

完成态不是终点。
A completion state is not an endpoint.
完成态是可交付状态。
A completion state is a deliverable state.

30.2 冻结

30.2 Freezing

冻结防止无休止修改。
Freezing prevents endless modification.
冻结提高系统可靠性。
Freezing increases system reliability.

30.3 版本切换

30.3 Version Switching

新版本不是修补旧版本。
New versions do not patch old versions.
新版本重定义边界。
New versions redefine boundaries.

30.4 工程终结语

30.4 Engineering Closing Statement

自我不是一次性工程。
The self is not a one-off project.
它是可版本化系统。
It is a versionable system.
演化的目标不是无限展开。
The goal of evolution is not infinite expansion.
而是可控更新。
It is controlled updating.